Comput Mech (2009) 44:745–755
DOI 10.1007/s00466-009-0406-3

ORIGINAL PAPER

Piecewise oblique boundary treatment for the elastic–plastic wave equation on a cartesian grid

Guido Giese

Received: 14 December 2008 / Accepted: 28 June 2009 / Published online: 12 July 2009

Abstract Numerical schemes for hyperbolic conservation laws in 2-D on a Cartesian grid usually have the advantage of being easy to implement and showing good computational performances, without allowing the simulation of "real-world" problems on arbitrarily shaped domains. In this paper a numerical treatment of boundary conditions for the elastic–plastic wave equation is developed, which allows the simulation of problems on an arbitrarily shaped physical domain surrounded by a piece-wise smooth boundary curve, but using a PDE solver on a rectangular Cartesian grid with the afore-mentioned advantages.

Keywords Elasticity · Plasticity · Elastic–plastic waves · Wave propagation in solids · Boundary conditions

1 Introduction

There exist a large number of numerical schemes for hyperbolic equations (cf. [3,10,11] or [24]), some of which have been used for the simulation of waves in solids (e.g. [12–17]). Even though most authors claim that their numerical scheme works on any grid—structured (e.g. Cartesian) or unstructured grid—most numerical examples presented are calculated on the most simple, i.e. a Cartesian grid in the shape of a rectangle in two dimensions (2-D) or a right parallelepiped in three dimensions (3-D), for the simple reason that an implementation of numerical schemes on arbitrary domains and the corresponding treatment of curved boundaries causes a high analytical and programming effort.

For practical applications, however, an implementation of numerical schemes for arbitrarily shaped domains with curved boundaries is essential. Numerical schemes using unstructured grids (e.g. with cells consisting of triangles approximating the shape of the curved boundary) do not only imply a higher implementation effort, but also result in a significantly higher memory usage and longer computation time—especially for the elastic–plastic wave equation, which is a system of fifteen equations/variables in 3-D and ten equations in 2-D.

For that reason, several Cartesian-boundary methods have been developed, especially in fluid dynamics (e.g. [2]). The main obstacles which have to be overcome when developing such Cartesian-boundary methods (especially in the presence of shocks) are stability and accuracy. In [1] a Cartesian-boundary method is developed where stability is achieved by sizing the domain of dependence for the individual Riemann problems according to the grid-spacing, introducing so-called h-boxes, whereas in [23] cut cells are merged with appropriate surrounding cells if the area of the cut cell is smaller than half the area of the uncut cells. In [21] cut cells are treated as whole cells, making an initial guess for the fluxes of such cells and then correcting these values to maintain conservation. All these methods are first-order accurate along the boundary. Although these methods are conceptually simple, they become technically rather complicated and computationally relatively expensive when being implemented.

To avoid these disadvantages, one can follow the Ansatz of Forrer in [6] who was able to simulate the Euler equations in 2-D on an arbitrarily shaped physical domain but using a structured rectangular Cartesian grid, by describing the physical boundary by a smooth curve separating inner cells (i.e. cells within the physical domain) and boundary cells (=ghost cells) in the rectangular numerical grid. Moreover, after each time-step the values of the physical variables in the ghost cells are set by "mirroring" each ghost cell at the

G. Giese (✉)
Seminar for Applied Mathematics, Swiss Federal Institute of Technology, Zurich, Switzerland
e-mail: guido_giese@yahoo.com

tangent of the curved boundary on a cell inside the boundary on the numerical grid (cf. below). For the elastic–plastic wave equation, which is considered in this paper, the implementation of a similar treatment for curved boundary conditions is more complicated—due to the relatively complex physical behavior of the equation and the different type of boundary conditions used in practical applications—but shows the same advantages as in fluid dynamics.

The wave equation of an elastic material represents a hyperbolic conservation law for the momentum and the strain variables and can be solved with standard methods for conservation laws. However, for solids which undergo plastic deformation, only the momentum but not the strain variables are conserved. Hence, in addition to flux calculations one has to solve the stress-strain relationship, which has the form of an ordinary differential equation (ODE) in every point (cf. [7]).

The purpose of this paper is to extend the numerical method developed in [7] called Method of Transport for purely rectangular computation domains to arbitrary domains and hence enabling the simulation of real-world problems. Thus, this paper develops a methodology to integrate piecewise oblique boundaries into the numerical treatment of the Method of Transport. Therefore, this paper starts with the general physical equations under consideration in 3-D and 2-D. Further, the Method of Transport used for the numerical computations in this paper is summarized. Details of the numerical scheme can be found in [4] and [5] for the Euler equations and in [7] for the elastic–plastic wave equation.

Afterwards, the implementation of curved boundary conditions on a rectangular Cartesian grid for the elastic–plastic wave equation is discussed in detail. It is important to mention that the boundary treatment presented in this paper is independent of the numerical scheme used for solving the hyperbolic equations.

2 Model equations

A standard formulation of the elastic–plastic wave equation is used in the following, which is based on the assumption of small strains, i.e. it is a linearization of the general flow equations (cf. [8,18]). Furthermore, the elastic–plastic wave equation is formulated as a first order system, which means that three types of physical variables appear: The symmetric stress tensor $\underline{\boldsymbol{\sigma}}$, the symmetric strain tensor $\underline{\boldsymbol{\epsilon}}$ and the velocity vector $\vec{\boldsymbol{v}}$. Furthermore, the deviatoric stress tensor $\underline{s}$ is defined as $s_{ij} = \sigma_{ij} - \frac{1}{3}\delta_{ij}\sigma_{kk}$. The system consists of equations describing the conservation of momentum, the compatibility relations between velocity and strain variables and the Prandtl model equations for describing the relationship between stress and strain (cf. [22]):

Conservation of momentum:

$$\frac{\partial}{\partial t}v_i = \frac{1}{\rho}\sum_{j=1}^{3}\frac{\partial}{\partial x_j}\sigma_{ij} \quad (\rho = \text{density}) \tag{1}$$

Compatibility relationship:

$$\frac{\partial}{\partial t}\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial}{\partial x_j}v_i + \frac{\partial}{\partial x_i}v_j\right) \quad i \le j = 1,\ldots,3 \tag{2}$$

Stress-strain relationship:

$$\frac{\partial}{\partial t}\epsilon_{ij} = \frac{1+\nu}{E}\frac{\partial}{\partial t}\sigma_{ij} - \frac{\nu}{E}\delta_{ij}\frac{d}{dt}\sigma_{kk} + s_{ij}\frac{d}{dt}\chi \tag{3}$$

with ν = Poisson's ratio, E = Young's modulus. For elasticity equation (3) reduces with $\frac{d}{dt}\chi \equiv 0$ to Hooke's law. In order to distinguish between elastic and plastic deformation the so-called von Mises yield function (cf. [20]) is used:

$$f(\underline{\boldsymbol{s}}) = \frac{1}{2}s_{ij}s_{ij} =: \kappa^2. \tag{4}$$

Basically, plasticity occurs in a certain point if the current function $\kappa(t)$ in that point attains the value of $\kappa_0(t)$, which is the maximal value of $\kappa(t)$ in the past, i.e. with

$$\kappa_0(t) = \max_{t_0 \le t' \le t}\kappa(t')$$

three different cases may occur:

1. $\kappa(t) < \kappa_0(t)$: Elastic deformation.
2. $\kappa(t) = \kappa_0(t)$ and $\frac{d}{dt}\kappa \le 0$: Elastic deformation.
3. $\kappa(t) = \kappa_0(t)$ and $\frac{d}{dt}\kappa > 0$: Plastic deformation.

This plasticity model reduces to the well know hysteresis-curve as shown in Fig. 1 in the case of one stress and one strain variable. For small stresses, i.e. $|\sigma| \le \kappa_0$ the relationship is linear (Hooke's law). However, if the stress $|\sigma|$ exceeds a certain value κ_0 then plastic flow occurs. Furthermore, unloading processes are always assumed to be elastic in our plasticity model. After the plastic loading and the elastic unloading process, plasticity will occur again if $|\sigma| \ge \kappa_1$ with κ_1 being the largest value of the stress $|\sigma|$ in the past.

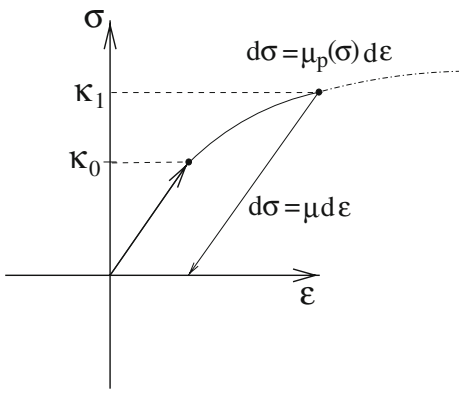

Fig. 1 Response of the strain variable ϵ to the stress σ in the case when hysteresis occurs

Furthermore, for this yield-criterion the function $\frac{d}{dt}\chi$ from Eq. (3) can be written in the form

$$\frac{d}{dt}\chi = \frac{1}{2\kappa}\left(\frac{1}{\mu_p(\kappa)} - \frac{1}{\mu}\right)\frac{d}{dt}\kappa \tag{5}$$

with a measured function $\mu_p(\kappa) \leq \mu$ (cf. Fig. 1) and the elastic shear modulus $\mu = \frac{E}{2(1+\nu)}$.

2.1 Model equation in 2-D

Without loss of generality the treatment of boundary conditions is based on a solid under the so-called plane strain condition in 2-D (cf. [18]), i.e. the z-component of the displacement and velocity vector is vanishing:

$$\epsilon_{13} = \epsilon_{23} = \epsilon_{33} = \sigma_{23} = \sigma_{13} = v_3 \equiv 0. \tag{6}$$

Hence, the flux Eqs. (1) and (2) reduce to the form:

$$\begin{pmatrix} v_1 \\ v_2 \\ \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ \epsilon_{12} \end{pmatrix}_t = \begin{pmatrix} \frac{\sigma_{11}}{\rho} \\ \frac{\sigma_{12}}{\rho} \\ v_1 \\ 0 \\ 0 \\ \frac{1}{2}v_2 \end{pmatrix}_x + \begin{pmatrix} \frac{\sigma_{12}}{\rho} \\ \frac{\sigma_{22}}{\rho} \\ 0 \\ v_2 \\ 0 \\ \frac{1}{2}v_1 \end{pmatrix}_y \tag{7}$$

Although ϵ_{33} is constant according to Eqs. (6) and (7), ϵ_{33} is left in the equation for the following ideas, because when investigating elastic–plastic waves one can replace strains with stresses and the corresponding stress component σ_{33} does not vanish.

The general stress-strain relationship contained in Eqs. (3) and (5) reduces under the assumption (6) to

$$\begin{pmatrix} \dot\epsilon_{11} \\ \dot\epsilon_{22} \\ \dot\epsilon_{33} \\ \dot\epsilon_{12} \end{pmatrix} = \underline{\mathcal{C}}(\underline{\boldsymbol{\sigma}}) \begin{pmatrix} \dot\sigma_{11} \\ \dot\sigma_{22} \\ \dot\sigma_{33} \\ \dot\sigma_{12} \end{pmatrix} \tag{8}$$

with $\underline{\mathcal{C}}(\underline{\boldsymbol{\sigma}})$

$$= \begin{pmatrix} a+\frac{h(\kappa)}{4\mu\kappa^2}s_{11}^2 & b+\frac{h(\kappa)}{4\mu\kappa^2}s_{11}s_{22} & b+\frac{h(\kappa)}{4\mu\kappa^2}s_{11}s_{33} & \frac{h(\kappa)}{2\mu\kappa^2}s_{11}\sigma_{12} \\ b+\frac{h(\kappa)}{4\mu\kappa^2}s_{22}s_{11} & a+\frac{h(\kappa)}{4\mu\kappa^2}s_{22}^2 & b+\frac{h(\kappa)}{4\mu\kappa^2}s_{22}s_{33} & \frac{h(\kappa)}{2\mu\kappa^2}s_{22}\sigma_{12} \\ b+\frac{h(\kappa)}{4\mu\kappa^2}s_{33}s_{11} & b+\frac{h(\kappa)}{4\mu\kappa^2}s_{33}s_{22} & a+\frac{h(\kappa)}{4\mu\kappa^2}s_{33}s_{33} & \frac{h(\kappa)}{2\mu\kappa^2}s_{33}\sigma_{12} \\ \frac{h(\kappa)}{4\mu\kappa^2}s_{11}\sigma_{12} & \frac{h(\kappa)}{4\mu\kappa^2}s_{22}\sigma_{12} & \frac{h(\kappa)}{4\mu\kappa^2}s_{33}\sigma_{12} & \frac{1}{2\mu}+\frac{h(\kappa)}{2\mu\kappa^2}\sigma_{12}^2 \end{pmatrix} \tag{9}$$

$$\begin{aligned} \kappa^2 &= \frac{1}{2}s_{ij}s_{ij} = \frac{1}{3}[(\sigma_{11}-\sigma_{33})^2 + (\sigma_{22}-\sigma_{33})^2 \\ &\quad -(\sigma_{11}-\sigma_{33})(\sigma_{22}-\sigma_{33})] + \sigma_{12}^2 \\ a &:= \frac{1}{9K} + \frac{1}{3\mu} \\ b &:= \frac{1}{9K} - \frac{1}{6\mu} \\ h(\kappa) &:= \frac{\mu}{\mu_p(\kappa)} - 1 \end{aligned} \tag{10}$$

with the Bulk modulus $K = \frac{E}{3(1-2\nu)}$.

The following definitions are used for plane strain:

$$\boldsymbol{V} = (v_1, v_2, \epsilon_{11}, \epsilon_{22}, \epsilon_{33}, \epsilon_{12})^T \tag{11}$$

$$\boldsymbol{U} = (v_1, v_2, \sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12})^T \tag{12}$$

with which the 2-D system can be written in the form

$$\boldsymbol{V}_t + \nabla \cdot \underline{\boldsymbol{L}}(\boldsymbol{U}) = 0 \tag{13}$$

$$(\epsilon_{11}, \epsilon_{22}, \epsilon_{33}, \epsilon_{12})_t^T = \underline{\mathcal{C}}(\underline{\boldsymbol{\sigma}})(\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12})_t^T \tag{14}$$

Equation (13) summarizes the conservation of momentum and the compatibility relations (7) ($\underline{\boldsymbol{L}}(\boldsymbol{U})$ is a linear function of $\boldsymbol{U}$, since both types of equations are linear) and Eq. (14) contains the stress-strain relationship from (8).

It is noteworthy that the system (13), (14) is not (cannot be written as) a pure hyperbolic conservation law, unless the material is purely elastic where the matrix $\underline{\mathcal{C}}$ is constant. Furthermore, the stress-strain relationship is used in the following form

$$\boldsymbol{U}_t = \underline{\mathcal{D}}(\underline{\boldsymbol{\sigma}})\boldsymbol{V}_t \tag{15}$$

where the existence of the matrix $\underline{\mathcal{D}}(\underline{\boldsymbol{\sigma}})$ can be seen from Eq. (8).

For elastic waves, i.e. $h(\kappa) \equiv 0$ the wave equation reduces to the closed form:

$$\begin{pmatrix} v_1 \\ v_2 \\ \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \end{pmatrix}_t = \nabla \cdot \begin{pmatrix} \frac{1}{\rho}\sigma_{11} & \frac{1}{\rho}\sigma_{12} \\ \frac{1}{\rho}\sigma_{12} & \frac{1}{\rho}\sigma_{22} \\ v_1(K+\frac{4}{3}\mu) & v_2(K-\frac{2}{3}\mu) \\ v_1(K-\frac{2}{3}\mu) & v_2(K+\frac{4}{3}\mu) \\ v_1(K-\frac{2}{3}\mu) & v_2(K-\frac{2}{3}\mu) \\ v_2\mu & v_1\mu \end{pmatrix} \tag{16}$$

2.2 The method of transport

The basic idea of the Method of Transport which is used for numerical simulations in this paper is to rewrite the flux equation (13) equivalently as a coupled system of advection equations. Therefore, one can introduce a set of direction vectors $\vec{\boldsymbol{n}}_i$, $i = 1, \ldots, k$, not necessarily of unit length, which

have to fulfill the following two consistency relations:

$$\sum_{i=1}^{k} \vec{n}_i = 0, \quad \frac{1}{k}\sum_{i=1}^{k} \vec{n}_i \vec{n}_i^T = \underline{I} \tag{17}$$

With these definitions one can rewrite the flux equation (13) (or 7) as follows:

$$\boldsymbol{V}_t + \quad \nabla \cdot c\underline{\boldsymbol{L}} = 0 \tag{18}$$

$$\iff \quad \frac{1}{k}\sum_{i=1}^{k} \left\{ (\boldsymbol{V} + \underline{\boldsymbol{L}}\vec{n}_i)_t + \nabla \cdot c(\boldsymbol{V} + \underline{\boldsymbol{L}}\vec{n}_i)\vec{n}_i^T \right\} = 0$$

$$\iff \quad \sum_{i=1}^{k} \{(\boldsymbol{R}_i)_t + \nabla \cdot c\boldsymbol{R}_i \vec{n}_i^T\} = 0 \tag{19}$$

with the identity matrix $\underline{I}$ and the quantities $\boldsymbol{R}_i = \frac{1}{k}(\boldsymbol{V} + \underline{\boldsymbol{L}}\vec{n}_i)$. One observes that the flux equation can be written as a system of coupled advection equations, each of which transports the quantity $\boldsymbol{R}_i$ at the velocity $c\vec{n}_i$.

Note that Eq. (19) is strictly equivalent to the original equation (18). The numerical approximation consists of decoupling the system, i.e. at a certain time-step t^n one defines the independent quantities

$$\boldsymbol{R}_i(\vec{x}, t^n) := \frac{1}{k}\left(\boldsymbol{V}(\vec{x}, t^n) + \underline{\boldsymbol{L}}(\vec{x}, t^n)\vec{n}_i\right) \tag{20}$$

to solve the advection equations

$$(\boldsymbol{R}_i)_t + \nabla \cdot \left(\boldsymbol{R}_i c\vec{n}_i^T\right) = 0 \quad \forall i \tag{21}$$

independently, i.e. one calculates the quantities $\boldsymbol{R}_i(\vec{x}, t)$ on the time interval $t \in [t^n, t^{n+1}]$ using Eq. (21) given the initial solution $\boldsymbol{R}_i(\vec{x}, t^n)$. At the next time-step t^{n+1} the update for the vector $\boldsymbol{V}$ reads:

$$\boldsymbol{V}(\vec{x}, t^{n+1}) = \sum_{i=1}^{k} \boldsymbol{R}_i(\vec{x}, t^{n+1}) \tag{22}$$

This yields a first order approximation for the exact solution in time if the consistency relations (17) hold, since then

$$\boldsymbol{V}_t = \sum_{i=1}^{k} (\boldsymbol{R}_i)_t = -\sum_{i=1}^{k} \nabla \cdot (c\boldsymbol{R}_i \vec{n}_i^T) = -\nabla \cdot c\underline{\boldsymbol{L}}.$$

and consequently

$$\boldsymbol{V}(\vec{x}, t^n + \Delta t) - \sum_{i=1}^{k} \boldsymbol{R}_i(\vec{x}, t^n + \Delta t) = \mathcal{O}(\Delta t^2)$$

For approximations of higher order in time one has to add correction terms into the flux equations (cf. [7,19]).

There is an infinite number of possible propagation vectors that fulfill the consistency relation (17), e.g. one can simply use the four diagonal directions $\vec{n}_i = (\pm 1, \pm 1)^T$.

So far the discussion has been semi-discrete only, i.e. the space variables still need to be discretized. The space discretization can be done as for all types of finite volume methods, i.e. one has to add the following two steps:

1. At a certain time-level one computes cell-averages for each cell I_{ij} for the quantity $\boldsymbol{V}$ according to Eq. (22):

$$\bar{\boldsymbol{V}}_{ij}^{n+1} = \frac{1}{|I_{ij}|}\sum_{l=1}^{k} \int_{I_{ij}} \boldsymbol{R}_l(\vec{y}, t^{n+1}) d\vec{y} \tag{23}$$

2. Before each time-step one has to reconstruct the function $\boldsymbol{V}(\vec{x}, t)$ by polynomials in space from the previously updated cell-averages.

The Method of Transport only yields an update of $\boldsymbol{V}$, i.e. the velocity and strain variables. The stress variables $\underline{\sigma}$ have to be updated by integrating the ODE (8) in time, which makes it necessary to reconstruct the behavior of $\underline{\epsilon}(t)$ on the time-interval $[t^n, t^{n+1}]$ by polynomials (for details cf. [7]).

3 Boundary conditions

In fluid dynamics (e.g. when simulating the Euler equations) the physical flow is typically surrounded by boundaries where the flows' velocity vector has to be tangent to the boundary curve, which can be achieved as follows for the Euler equations (cf. [6]): The density, the energy and the tangent component of the velocity are "copied" from the inner cell into the ghost cell (i.e. boundary cell), whereas the normal velocity in the ghost cell is minus the value in the inner cell.

For elastic–plastic deformation in solids, however, the physical behavior and hence the boundary conditions engineers want to impose in their simulations of applied problems are quite different. The two most commonly used boundary conditions applied to the boundary of the physical domain in practical problems are (with $\vec{n}$ denoting the normal of the boundary curve ∂G of the physical domain G in 2-D):

1. Velocity boundary conditions:

$$\vec{v}|_{\partial G} = \vec{v}_G \tag{24}$$

   For example, one can assume that the solid is fixed at the boundary so that the particles at the boundary cannot move and hence $\vec{v}|_{\partial G} = 0$.
2. Stress boundary conditions, typically the normal traction $\vec{F}_n$ on the boundary is prescribed:

$$\underline{\sigma}\vec{n} = \vec{F}_n \tag{25}$$

   The special case $\vec{F}_n = 0$ is the case of a free boundary.

In principle, mixed velocity and stress conditions are possible as well and can be treated in the same way as presented below. To ensure stability (cf. [9]), the concept of Riemann-type boundary conditions is used, i.e. in 2-D one can prescribe two physical variables on the boundary (since there are two positive eigenvalues of the Jacobian of the flux), the other variables have to be computed consistently (see below). Thus, after each time step the values for the velocity, the stress and the strain variables are computed in the ghost cells of the numerical grid representing the boundary so that they simulate the conditions (24) or (25) or a mixture of them.

The following analysis starts with such boundary conditions in the one dimensional (1-D) case, where one has to find the correct values for the physical variables at the end of a 1-D solid bar and then the results are extended to 2-D, where the physical domain is surrounded by an arbitrary piece-wise smooth curve.

3.1 Boundary conditions in 1-D

To understand the problem of prescribing boundary conditions one can start with the 1-D problem, i.e. one considers a long solid bar to prescribe a certain physical situation (e.g. a certain stress) at the end of the bar to start or absorb waves.

Apart from the grid cells representing the bar, one has to use a ghost-cell connected to the end of the physical domain to prescribe the values of the physical variables one wants to prescribe at the boundary when using finite-volume schemes. Hence, between the ghost cell with state $\boldsymbol{U}^g$ and the first inner cell $\boldsymbol{U}^i$ one has to solve the corresponding Riemann problem (for details cf. below) to find out how to set the vector $\boldsymbol{U}^g$ to produce the desired stress waves propagating into the physical domain (cf. Fig. 2).

In the most general case the solution of the Riemann problem consists of four elastic contact discontinuities

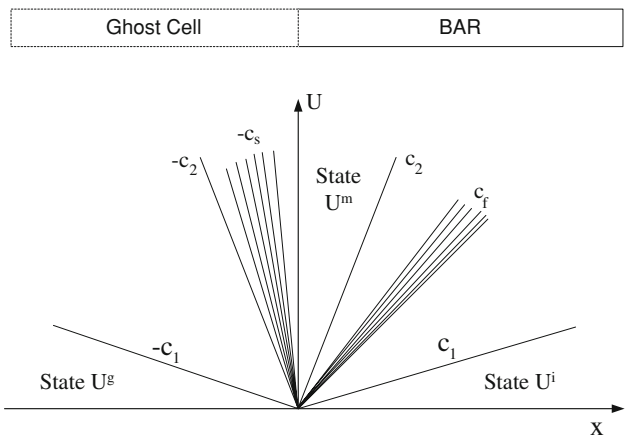

Fig. 2 Example for the Riemann problem between the ghost-cell in state $\boldsymbol{U}^g$ and the state in the first inner cell $\boldsymbol{U}^i$

propagating at $\pm c_1$ (longitudinal wave) and $\pm c_2$ (transverse wave) and four plastic rarefaction waves travelling at $\pm c_f(\underline{\boldsymbol{\sigma}})$ and $\pm c_s(\underline{\boldsymbol{\sigma}})$ (fast and slow plastic wave). $\boldsymbol{U}^m$ is the middle state at $x = 0$. For imposing boundary conditions, the value of $\boldsymbol{U}^g$ has to be set so that the state $\boldsymbol{U}^i$ contains the variable values that are physically applied to the end of the bar. Therefore, it is obviously enough to consider a Riemann-solution consisting of positive wave-speeds only, i.e. considering the special case $\boldsymbol{U}^g = \boldsymbol{U}^m$.

Hence, one has to be able to solve the Riemann problem between two states, which is demonstrated for elastic waves first and then for plastic waves in the following.

3.1.1 *Riemann problem for elastic waves*

As explained in [11], for the linear elastic case the two vectors $\boldsymbol{U}^i$ and $\boldsymbol{U}^g$ are connected by a linear combination of eigenvectors $\boldsymbol{r}_k$ of the Jacobian matrix $\underline{\boldsymbol{A}}_{el}$ of the flux of the purely elastic wave equation along the x-axis indicated in Eq. (16):

$$\boldsymbol{U}^i - \boldsymbol{U}^g = \sum_k \alpha_k \boldsymbol{r}_k \tag{26}$$

where the corresponding eigenvectors and eigenvalues are given in Table 1.

Since there are two positive eigenvalues transporting two independent right eigenvectors, one can prescribe two independent physical variables on the boundary to come to stable boundary conditions (Riemann-type boundary conditions) – all other variables have to be determined consistently by solving the Riemann problem, i.e. one has to determine the two values of the α_i in Eq. (26) for the two eigenvectors corresponding to the two positive eigenvalues c_1 and c_2.

Hence, for the afore-mentioned cases (velocity or stress boundary conditions or mixed conditions) one can then choose the values in the boundary cell, which is demonstrated in the following two examples (the indices g and i mean the values in the boundary=ghost cell and the inner cell):

1. Stress boundary conditions with $\vec{\boldsymbol{F}}_n \equiv 0$: Since the force has to vanish on the boundary, Eq. (25) forces us to prescribe $\sigma^g_{11} = \sigma^g_{12} = 0$, since the normal vector $\vec{\boldsymbol{n}}$ is parallel to the x-axis. Equation (26) implies:

$$v_1^g = v_1^i - \frac{\sigma^g_{11} - \sigma^i_{11}}{\rho c_1}$$
$$v_2^g = v_2^i - \frac{\sigma^g_{12} - \sigma^i_{12}}{\rho c_2}$$
$$\sigma^g_{22} = \sigma^i_{22} - \frac{K - 2/3\mu}{c_1}(v_1^g - v_1^i)$$
$$\sigma^g_{33} = \sigma^i_{33} - \frac{K - 2/3\mu}{c_1}(v_1^g - v_1^i)$$

Table 1 Decomposition of the 1-D elastic wave equation into right eigenvectors $\boldsymbol{r}_i$ along the x- and y-axis

Eigenvalue	$\boldsymbol{r}_i$ in x-direction	$\boldsymbol{r}_i$ in y-direction
$\pm c_1$	$\begin{pmatrix} \frac{1}{2}(v_1 \pm \frac{\sigma_{11}}{\rho c_1}) \\ 0 \\ \frac{1}{2}(\pm v_1 \rho c_1 + \sigma_{11}) \\ \frac{1}{2}(\pm v_1 \rho c_1 + \sigma_{11})\frac{K-2/3\mu}{K+4/3\mu}) \\ \frac{1}{2}(\pm v_1 \rho c_1 + \sigma_{11})\frac{K-2/3\mu}{K+4/3\mu}) \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ \frac{1}{2}(v_2 \pm \frac{\sigma_{22}}{\rho c_1}) \\ \frac{1}{2}(\pm v_2 \rho c_1 + \sigma_{22})\frac{K-2/3\mu}{K+4/3\mu}) \\ \frac{1}{2}(\pm v_2 \rho c_1 + \sigma_{22}) \\ \frac{1}{2}(\pm v_2 \rho c_1 + \sigma_{22})\frac{K-2/3\mu}{K+4/3\mu}) \\ 0 \end{pmatrix}$
$\pm c_2$	$\begin{pmatrix} 0 \\ \frac{v_2}{2} \pm \frac{\sigma_{12}}{2\rho c_2} \\ 0 \\ 0 \\ 0 \\ \frac{\sigma_{12}}{2} \pm \frac{v_2 \rho c_2}{2} \end{pmatrix}$	$\begin{pmatrix} \frac{v_1}{2} \pm \frac{\sigma_{12}}{2\rho c_2} \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{\sigma_{12}}{2} \pm \frac{v_1 \rho c_2}{2} \end{pmatrix}$
0	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ \sigma_{22} - \frac{K-2/3\mu}{K+4/3\mu}\sigma_{11} \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ \sigma_{11} - \frac{K-2/3\mu}{K+4/3\mu}\sigma_{22} \\ 0 \\ 0 \\ 0 \end{pmatrix}$
0	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \sigma_{33} - \frac{K-2/3\mu}{K+4/3\mu}\sigma_{11} \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \sigma_{33} - \frac{K-2/3\mu}{K+4/3\mu}\sigma_{22} \\ 0 \end{pmatrix}$

2. Velocity boundary condition with $\vec{\boldsymbol{v}}_g \equiv 0$, i.e. one prescribes the velocity $v_1^g = v_2^g = 0$. From the Rankine-Hugoniot jump conditions for Eq. (16) we get:

$$\sigma_{11}^g = \sigma_{11}^i - \frac{K + 4/3\mu}{c_1}(v_1^g - v_1^i)$$
$$\sigma_{22}^g = \sigma_{22}^i - \frac{K - 2/3\mu}{c_1}(v_1^g - v_1^i)$$
$$\sigma_{33}^g = \sigma_{33}^i - \frac{K - 2/3\mu}{c_1}(v_1^g - v_1^i)$$
$$\sigma_{12}^g = \sigma_{12}^i - \frac{\mu}{c_2}(v_2^g - v_2^i)$$

Mixed velocity/stress boundary conditions can be treated analogously.

3.1.2 Riemann problem for plastic waves

For solving the Riemann problem for plasticity, one can rewrite the system in the 1-D case (i.e. all y-dependencies are neglected) by plugging equation (8) into (7) to obtain (cf. [25])

$$\underline{\boldsymbol{\mathcal{A}}}(\underline{\boldsymbol{\sigma}})\boldsymbol{U}_t = \underline{\boldsymbol{A}}\boldsymbol{U}_x \tag{27}$$

with

$$\underline{\boldsymbol{\mathcal{A}}}(\underline{\boldsymbol{\sigma}}) = \left(\begin{array}{cc|c} 1 & 0 & 0\ldots \\ 0 & 1 & 0\ldots \\ \hline 0 & 0 & \\ \vdots & \vdots & \underline{\boldsymbol{\mathcal{C}}}(\underline{\boldsymbol{\sigma}}) \end{array}\right) = \underline{\boldsymbol{\mathcal{D}}}(\underline{\boldsymbol{\sigma}})^{-1}$$
$$\underline{\boldsymbol{A}} = \begin{pmatrix} 0 & 0 & \frac{1}{\rho} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \end{pmatrix} \tag{28}$$

The operator $\frac{d}{dt}$ in the characteristic surface is defined as:

$$\frac{d}{dt}\boldsymbol{U} = \boldsymbol{U}_t + c\boldsymbol{U}_x$$

Then, one can rewrite equation (27) in the form

$$\underline{\boldsymbol{\mathcal{A}}}(\underline{\boldsymbol{\sigma}})\left(\frac{d\boldsymbol{U}}{dt} - c\boldsymbol{U}_x\right) - \underline{\boldsymbol{A}}\boldsymbol{U}_x = 0 \tag{29}$$

One has too look for a linear combination of these equations containing operators of the form $\frac{d}{dt}$ only. Therefore, one can

multiply equation (29) from the left with a vector $\boldsymbol{l}$. Hence, $\boldsymbol{l}$ has to fulfill the following condition

$$\boldsymbol{l}^T \underline{\boldsymbol{Q}}(c) = 0, \quad \text{with } \underline{\boldsymbol{Q}}(c) := -c\underline{\boldsymbol{\mathcal{A}}} - \underline{\boldsymbol{A}} \tag{30}$$

which means that $\boldsymbol{l}$ is a left eigenvector of $\underline{\boldsymbol{Q}}$ with eigenvalue zero. Thus, one has to find the the roots of the characteristic polynomial $\chi(c) := \det \underline{\boldsymbol{Q}}(c)$ which turns out to be

$$\chi(c)/c^2 = \frac{\left(\alpha\, K\kappa^2 - 3\, K\alpha\, {\sigma_{12}}^2 + 3\, K\kappa^2 + 4\, {c_2}^2\kappa^2 + \alpha\, {c_2}^2\Sigma^2\right)}{{c_2}^2} - \frac{\left(\alpha\, K\kappa^2 + 7\, {c_2}^2\kappa^2 + \alpha\, {c_2}^2\Sigma^2 + \alpha\, {c_2}^2\kappa^2 + {c_2}^2\alpha\, {\sigma_{12}}^2 + 3\, K\kappa^2\right)}{{c_2}^4} c^2 + \frac{\kappa^2\,(3+\alpha)}{{c_2}^4} c^4$$

with

$$\Sigma^2 := (\sigma_{11} - \sigma_{22})^2, \quad \alpha := \frac{1}{\mu_p(\kappa)} - \frac{1}{\mu}$$

Besides the double eigenvalue 0 there are four characteristic speeds $\pm c_1$ and $\pm c_2$ in the elastic case with

$$c_1 = \sqrt{\frac{K + 4/3\mu}{\rho}}$$

$$c_2 = \sqrt{\frac{\mu}{\rho}}$$

and four characteristic speeds $\pm c_s(\underline{\boldsymbol{\sigma}})$ (slow plastic wave) and $\pm c_f(\underline{\boldsymbol{\sigma}})$ (fast plastic wave) for plastic deformation, which satisfy

$$0 < c_s(\underline{\boldsymbol{\sigma}}) < c_2 < c_f(\underline{\boldsymbol{\sigma}}) < c_1.$$

As before, the velocity c_1 denotes the propagation of elastic longitudinal waves, whereas elastic transverse waves propagate at c_2. The left eigenvectors fulfilling condition (30) for the ("plastic") eigenvalues $\pm c_f(\underline{\boldsymbol{\sigma}})$ and $\pm c_s(\underline{\boldsymbol{\sigma}})$ can be written in the form (cf. [25])

$$\boldsymbol{l} := (\Psi,\, 1,\, \rho c\Psi,\, \Phi_a,\, \Phi_b,\, 2\rho c)^T. \tag{31}$$

The functions Ψ, Φ_a and Φ_b are somewhat complicated:

$$\begin{aligned}\Psi = {} & (-{c_2}^4\alpha\,\Sigma^2 + 4\,{c_2}^4\kappa^2 + 3\,K{c_2}^2\kappa^2 + K{c_2}^2\alpha\,\kappa^2 \\ & -3\,K{c_2}^2\alpha\,{\sigma_{12}}^2 - 3\,c^2K\kappa^2 - 4\,c^2{c_2}^2\alpha\,{\sigma_{12}}^2 \\ & -4\,c^2{c_2}^2\kappa^2 - c^2\alpha\,K\kappa^2 - c^2\alpha\,{c_2}^2\Sigma^2) \\ & /((-2\,\sigma_{11} + \sigma_{33} + \sigma_{22})c^2\sigma_{12}\,\alpha\,{c_2}^2) \\ \Phi_a = {} & (2\,{c_2}^4\kappa^2 - {c_2}^4\alpha\,\sigma_{22}\,\sigma_{33} - {c_2}^4\alpha\,\sigma_{33}\,\sigma_{11} + {c_2}^4\alpha\,\sigma_{22}\,\sigma_{11} \\ & + {c_2}^4\alpha\,{\sigma_{33}}^2 - 2\,c^2{c_2}^2\kappa^2 + c^2\alpha\,\sigma_{22}\,\sigma_{33}\,{c_2}^2 + c^2\alpha\,K\kappa^2 \\ & - 2\,c^2{c_2}^2\alpha\,{\sigma_{12}}^2 + c^2\alpha\,\sigma_{33}\,\sigma_{11}\,{c_2}^2 - c^2\alpha\,\sigma_{22}\,\sigma_{11}\,{c_2}^2 \\ & - c^2\alpha\,{\sigma_{33}}^2{c_2}^2 + 3\,c^2K\kappa^2 - 3\,K{c_2}^2\kappa^2 - K{c_2}^2\alpha\,\kappa^2 \\ & + 3\,K{c_2}^2\alpha\,{\sigma_{12}}^2)/(\sigma_{12}\,\alpha\,c(-2\,\sigma_{11} + \sigma_{33} + \sigma_{22}){c_2}^2)\end{aligned}$$

$$\begin{aligned}\Phi_b = {} & (c^2\alpha\,\sigma_{22}\,\sigma_{11}\,{c_2}^2 - c^2\alpha\,\sigma_{33}\,\sigma_{11}\,{c_2}^2 + c^2\alpha\,K\kappa^2 \\ & + \sigma_{33}\,{c_2}^4\alpha\,\sigma_{11} - c^2\alpha\,{c_2}^2\Sigma^2 + c^2\alpha\,{\sigma_{33}}^2{c_2}^2 \\ & - 3\,K{c_2}^2\kappa^2 - K{c_2}^2\alpha\,\kappa^2 + 3\,K{c_2}^2\alpha\,{\sigma_{12}}^2 + {c_2}^4\alpha\,\Sigma^2 \\ & - {c_2}^4\alpha\,{\sigma_{33}}^2 + {c_2}^4\alpha\,\sigma_{22}\,\sigma_{33} + 2\,{c_2}^4\kappa^2 - {c_2}^4\sigma_{22}\,\alpha\,\sigma_{11} \\ & - c^2{c_2}^2\alpha\,\sigma_{22}\,\sigma_{33} - 2\,c^2{c_2}^2\kappa^2 + 3\,c^2K\kappa^2 \\ & - 2\,{c_2}^2c^2\alpha\,{\sigma_{12}}^2)/(\sigma_{12}\,(-2\,\sigma_{11} + \sigma_{33} + \sigma_{22}){c_2}^2c\,\alpha)\end{aligned}$$

The corresponding compatibility relation along the characteristic $dx = \pm c_{s/f}(\underline{\boldsymbol{\sigma}})dt$ can be obtained from (29) with (30):

$$\boldsymbol{l}^T \underline{\boldsymbol{\mathcal{A}}}(\underline{\boldsymbol{\sigma}})d\boldsymbol{U} = 0. \tag{32}$$

As in the elastic case, one can restrict oneselves to the four positive eigenvalues $c_s(\underline{\boldsymbol{\sigma}}) < c_2 < c_f(\underline{\boldsymbol{\sigma}}) < c_1$ when calculating the matching boundary vector $\boldsymbol{U}^g$ depending on the inner state $\boldsymbol{U}^i$. The problem is that these four waves imply the existence of three intermediate states $\boldsymbol{U}^a$, $\boldsymbol{U}^b$ and $\boldsymbol{U}^c$, which have to be found by an iteration starting from the inner state $\boldsymbol{U}^i$. The numerical algorithm consists of the following four steps:

1. Applying Rankine-Hugoniot jump conditions for the elastic eigenvalue c_1 to come to the state $\boldsymbol{U}^a = \boldsymbol{U}^i + \beta_1\boldsymbol{r}_1$, where $\boldsymbol{r}_1$ is the eigenvector from Table 1 and
2. Connecting the two middle states $\boldsymbol{U}^a$ and $\boldsymbol{U}^b$ by solving the compatibility ODE (32) for the eigenvalue $c_f(\underline{\boldsymbol{\sigma}})$ from the characteristic speed $c_f(\underline{\boldsymbol{\sigma}}^a)$ to the speed $c_f(\underline{\boldsymbol{\sigma}}^b) = c_f(\underline{\boldsymbol{\sigma}}^a) - \beta_2$ and
3. Connecting $\boldsymbol{U}^b$ to $\boldsymbol{U}^c$ by Rankine-Hugoniot conditions for the eigenvalue c_2 by using the corresponding right eigenvector from Table 1, i.e. $\boldsymbol{U}^c = \boldsymbol{U}^b + \beta_3\boldsymbol{r}_2$ and finally
4. Solving the ODE (32) for the slow plastic eigenvalue $c_s(\underline{\boldsymbol{\sigma}})$ from the characteristic speed $c_s(\underline{\boldsymbol{\sigma}}^c)$ to $c_s(\underline{\boldsymbol{\sigma}}^g) = c_s(\underline{\boldsymbol{\sigma}}^c) - \beta_4$ to come to the boundary state $\boldsymbol{U}^g$

in each iteration step—cf. Fig. 3.

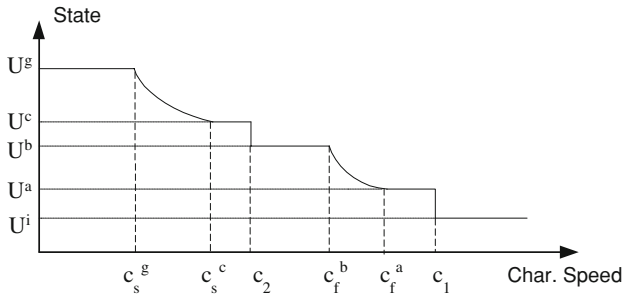

Fig. 3 Riemann solution (for the stress $\sigma_{11} + \sigma_{12}$) with positive eigenvalues only, including all four waves and three intermediate states a, b and c that have to be found by iteration to find the correct state $\boldsymbol{U}^g$ in the ghost cell

The iteration, which is based on Newton's method for the parameter vector $(\beta_1, \ldots, \beta_4)$ assures that after the iteration the vector $\boldsymbol{U}^g$ contains the stress and velocity components prescribed at the boundary and the consistently calculated values for the other stress-components and velocities. Even though there are four positive eigenvalues carrying four right eigenvectors in the plastic case, the number of independent boundary conditions that can be prescribed remains two, since there are two additional constraints to fulfill in the iteration: The two transitions from elastic wave to plastic wave (i.e. from c_1 to c_f and c_2 to c_s) have to occur on the yield surface, which constitutes a scalar constraint for each of the two transition points.

After the vector $\boldsymbol{U}^g$ containing the velocity and stress components in the ghost cell has been found, one can calculate the strains $\underline{\boldsymbol{\epsilon}}^g$ in the boundary cell by integrating the stress-strain relationship (14) from the stress state $\underline{\boldsymbol{\sigma}}^i$ in the inner cell to the stress state $\underline{\boldsymbol{\sigma}}^g$ in the ghost cell with a standard ODE-solver, e.g. a Runke-Kutta method. Since the right-hand side of the ODE might be discontinuous at the transition from elasticity to plasticity, one has to find the exact transition point in the stress-space by iteration and restart the ODE-solver (for details cf. [7]).

3.2 Boundary treatment on cartesian grids in 2-D

In the general case when simulating a domain in 2-D surrounded by a curved boundary instead of a simple 1-D boundary condition, one has to be able to prescribe boundary conditions along any curved boundary line.

The following boundary treatment is based on the ideas presented in [6] by Forrer who developed a method for treating curved boundaries on a Cartesian grid for the Euler equations. The basic idea is to compute cell updates for the state vector $\boldsymbol{U}_{ij}$ in each cell (i, j) on a Cartesian grid. However, after each time-step the values in the boundary cells are found by mirroring the cell at the tangent vector of the boundary curve into the physical domain (cf. Fig. 4) and solving a 1-D Riemann problem between the physical state found in the mirrored cell (which lies inside the computational domain) and the boundary cell, in which one can prescribe two physical values as in 1-D.

The boundary cells are now those cells that are right at or behind the curve describing the shape of the domain. The state $\boldsymbol{U}^g$ that has to be set in the boundary cell is found by solving a 1-D Riemann problem between this boundary cell and the cell's "mirror" cell in the physical domain similar to the 1-D case, i.e. the numerical algorithm consists of three steps:

1. One calculates the normal $\vec{\boldsymbol{n}}$ to the boundary curve pointing towards the middle of the ghost cell and then

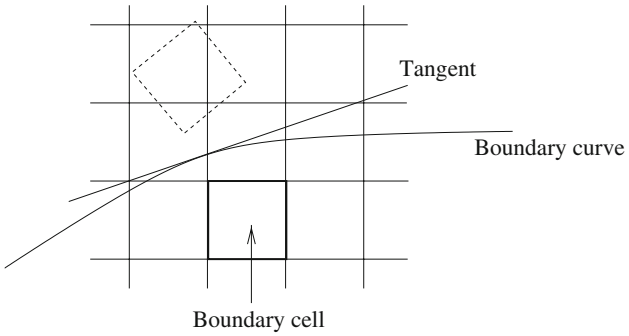

Fig. 4 The boundary cell (*bold line*) is mirrored in the computational domain (*dashed line*). Hence, the physical values in the mirrored state are known since it lies within the physical domain. The values in the boundary cell are then determined as in 1-D, i.e. by prescribing two independent physical variables in the boundary cell and then calculating the remaining physical variables by solving a 1-D Riemann problem between the boundary cell and the mirrored cell

2. Projects the physical equations to a 1-D problem along this direction $\vec{\boldsymbol{n}}$ and then
3. Calculates the boundary state $\boldsymbol{U}^g$ by solving the 1-D Riemann problem along the direction $\vec{\boldsymbol{n}}$ as in the 1-D case explained above, where (as in 1-D) one can prescribe two of the six physical variables contained in $\boldsymbol{U}^g$ as boundary conditions and the remaining four variables are found through the Riemann solution. The strains are found by integrating the stress-strain relationship (cf. Eq. 14).

Instead of projecting the physical equations into the direction $\vec{\boldsymbol{n}}$ of the normal vector to the boundary curve, one can rotate the coordinate frame by an angle φ so that in the new frame one has $\tilde{\vec{\boldsymbol{n}}} = (1, 0)$ and hence one can consider a 1-D Riemann problem along the x-axis. Therefore, the state vector $\boldsymbol{U}^i$ in the mirrored inner cell (as well as the state in the ghost cell $\boldsymbol{U}^g$) has to be transformed by a linear transformation:

$$\tilde{\boldsymbol{U}}^i = \underline{\boldsymbol{\Lambda}}_{\varphi} \boldsymbol{U}^i$$

Then, $\tilde{\boldsymbol{U}}^g$ can be computed as explained above by solving a Riemann problem along the x-axis between the two states $\tilde{\boldsymbol{U}}^g$ and $\tilde{\boldsymbol{U}}^i$. Afterwards, one uses the backward transformation to calculate the state vector in the ghost cell:

$$\boldsymbol{U}^g = \underline{\boldsymbol{\Lambda}}_{(-\varphi)} \tilde{\boldsymbol{U}}^g$$

This approach will lead to more complicated results than for the Euler equations, because the transformation of the physical quantities with the matrix $\underline{\boldsymbol{\Lambda}}_{\varphi}$ which will be derived in the following is much more complicated: Coordinates and velocities transform as rank one tensors, whereas the stress tensor transforms as a rank two tensor under a rotation $\underline{\boldsymbol{R}}$:

$$\underline{\boldsymbol{R}} = \begin{pmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \underline{\boldsymbol{R}} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x\cos(\varphi) + y\sin(\varphi) \\ -x\sin(\varphi) + y\cos(\varphi) \end{pmatrix} \tag{33}$$

$$\begin{pmatrix} v_1' \\ v_2' \end{pmatrix} = \underline{\boldsymbol{R}} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} v_1\cos(\varphi) + v_2\sin(\varphi) \\ -v_1\sin(\varphi) + v_2\cos(\varphi) \end{pmatrix} \tag{34}$$

$$\begin{pmatrix} \sigma_{11}' & \sigma_{12}' \\ \sigma_{12}' & \sigma_{22}' \end{pmatrix} = \underline{\boldsymbol{R}} \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{pmatrix} \underline{\boldsymbol{R}}^T$$

$$= \begin{pmatrix} \sigma_{11}\cos^2(\varphi) + \sigma_{22}\sin^2(\varphi) + 2\sigma_{12}\sin(\varphi)\cos(\varphi) & (\sigma_{22} - \sigma_{11})\sin(\varphi)\cos(\varphi) - \sigma_{12}(1 - 2\cos^2(\varphi)) \\ (\sigma_{22} - \sigma_{11})\sin(\varphi)\cos(\varphi) - \sigma_{12}(1 - 2\cos^2(\varphi)) & \sigma_{11}\sin^2(\varphi) + \sigma_{22}\cos^2(\varphi) - 2\sigma_{12}\sin(\varphi)\cos(\varphi) \end{pmatrix} \tag{35}$$

Thus, one can find a linear transformation $\underline{\boldsymbol{\Lambda}}_\varphi$ which transforms the state vector $\boldsymbol{U}$ defined in (12) into the state vector $\boldsymbol{U}' = \underline{\boldsymbol{\Lambda}}_\varphi \boldsymbol{U}$ in the rotated frame according to the relations (34) and (35). After doing some algebra, one obtains $\underline{\boldsymbol{\Lambda}}_\varphi$

$$= \begin{pmatrix} \cos(\varphi) & \sin(\varphi) & 0 & 0 & 0 & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 & 0 & 0 & 0 \\ 0 & 0 & \cos^2(\varphi) & \sin^2(\varphi) & 0 & 2\sin(\varphi)\cos(\varphi) \\ 0 & 0 & \sin^2(\varphi) & \cos^2(\varphi) & 0 & -2\sin(\varphi)\cos(\varphi) \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -\sin(\varphi)\cos(\varphi) & \sin(\varphi)\cos(\varphi) & 0 & -\sin^2(\varphi) + \cos^2(\varphi) \end{pmatrix} \tag{36}$$

which completes the boundary treatment in 2-D.

4 Numerical example and order estimate

To demonstrate the effectiveness of the proposed approach, a circular hole (i.e. a free boundary) in a plate under the plane strain condition (cf. Fig. 5) is considered in the following. From one side a plane wave is approaching which is to be scattered at the hole.

The following boundary conditions (with H denoting the Heaviside function) are chosen:

$$\sigma_{22}(x, y = 0, t) = H(t)$$

All variables are assumed to vanish in the computational domain $y \geq 0$ at the beginning. Thus, one can expect a compression wave to propagate into the domain which is scattered at the hole.

For the exact solution the force on the boundary vanishes everywhere on the surface S of the hole:

$$\vec{\boldsymbol{F}}_n = \underline{\boldsymbol{\sigma}}\vec{\boldsymbol{n}} = 0$$

In the following numerical simulation the numerical error e is computed, i.e.

$$e = \oint_S \|\vec{\boldsymbol{F}}_n\|_2 ds. \tag{37}$$

The exact solution would fulfill $e = 0$. The simulations were performed with the Method of Transport (cf. [7]) with third order accuracy in space and time. Table 2 shows the error and numerical order p_i of the boundary treatment which is defined as

$$p_i = \ln_2\left(\frac{e_i}{e_{2i}}\right)$$

with e_i indicating the numerical error calculated according to Eq. (37) and i denoting the width (in number of cells) of the rectangular grid. Although the boundary approximation is only first order accurate, the numerical order seems to be between first and second order accurate in this numerical experiment, which is consistent to the numerical experiments for the Euler equations in [6]. This is due to the fact that a second order interior scheme was used, which is applied to

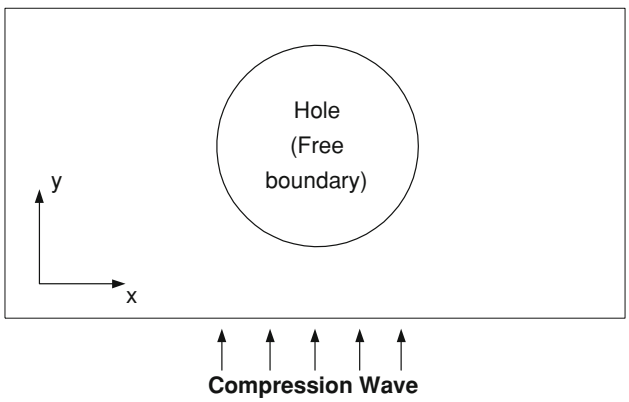

Fig. 5 Plate under plane strain with a circular hole, representing a free boundary (i.e. the force has to vanish on the surface of the hole.) At the lower boundary a compression wave is created propagating towards the hole

Table 2 Numerical errors and order for different grids

Grid used	Error e	Order p_i
50×25	0.0839	
100×50	0.0360	1.22
200×100	0.0134	1.43
400×200	0.00533	1.33
800×400	0.00207	1.36

The order of the boundary treatment is between one and two

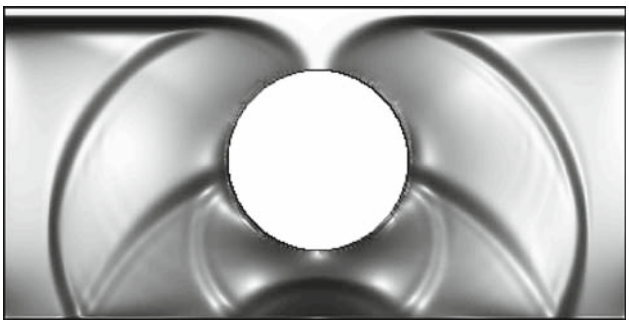

Fig. 6 Numerical solution of the stress component σ_{33} of a reflected contact discontinuity at the hole on a grid with 400×200 cells

more cells than the first order scheme for treating boundary cells and hence the numerical order appears to be greater than one.

The numerical simulation is shown in Fig. 6. The discontinuity which travels through the domain in the upper direction causes a reflected compression wave at the hole propagating in all directions. This reflection causes a von Schmidt Wave at the boundary which is not in the exact form of a cone, since the compression wave does not propagate parallel to the boundary. The von Schmidt wave itself causes a reflection at the hole which in its turn causes a second von Schmidt wave at the boundary and so forth.

5 Conclusion

It is quite obvious that simulating elastic plastic waves (or any other hyperbolic equation) on a domain that is described by curved boundaries usually implies an immense analytical and implementation effort, since the grid has to be adjusted to the curved boundary and furthermore, the computation is normally also slowed down due to the higher complexity of the grid.

The approach presented in this paper consists of using a rectangular Cartesian grid, where the implementation and also computations are usually fastest, and setting adequate boundary conditions within this Cartesian grid after every time-step of the PDE solver, where a 1-D projection of the physical equations onto the normal vector of the boundary curve is used to be able to calculate the correct physical values in the boundary cells by solving Riemann problems. Due to the higher complexity of the physical equations, the calculation of the boundary state is analytically and computationally much more complex than for the Euler equations. Nevertheless, the implementation is far more simple than using unstructured grids.

Compared to sophisticated numerical techniques using unstructured grids, the proposed methodology offers the clear advantage of being very simple and fast to implement and to yield good results for a fair effort. However, it has the disadvantage of being a first-order approximation only, due to the use of linearized boundaries to "mirror" cells, which clearly limits the accuracy of the methodology compared to grid-adjusted techniques. Consequently, users looking for the highest accuracy possible at any cost are advised to implement a higher-order boundary treatment.

The numerical examples show that the boundary conditions are at least first order accurate in time. To conclude, the method represents a good trade-off between accuracy on the one hand and implementation and computational effort on the other hand.

References

1. Berger MJ, LeVeque RJ (1991) A rotated difference scheme for Cartesian grids in complex geometries. AIAA Paper CP-91-1602
2. Clarke DK, Salas MD, Hassan HA (1986) Euler calculations for multielement airfoils using Cartesian grids. AIAA J 24:353–358
3. Colella P (1990) Multidimensional upwind methods for hyperbolic conservation laws. SIAM J Comp Phys 87:171–200
4. Fey M (1998) Multidimensional upwinding. Part I. The method of transport for solving the Euler equations. J Comput Phys 143: 159–180
5. Fey M (1998) Multidimensional upwinding. Part II. Decomposition of the Euler equations into advection equations. J Comput Phy 143:181–199
6. Forrer H, Jeltsch R (1998) A higher-order boundary treatment for cartesian-grid methods. J Comput Phys 140:259–277
7. Giese G, Fey M (2002) A genuinely multi-dimensional high-resolution scheme for the elastic–plastic wave equation. J Comput Phys 181:1–16
8. Gould PL (1983) Introduction to linear elasticity. Springer, Heidelberg
9. Kreiss H, Lorenz J (1989) Initial-boundary value problems and the Navier-Stokes equations. Academic Press, New York
10. Kröner D (1997) Numerical schemes for conservation laws. Wiley Teubner, New York
11. LeVeque RJ (1992) Numerical methods for conservation laws. Lectures in Mathematics. Birkhäuser, Boston
12. Lin X (1996) Numerical computation of stress waves in solids. Akademie Verlag, Berlin
13. Lin X, Ballmann J (1995) Improved bicharacteristic schemes for two-dimensional elastodynamic equations. Quart Appl Math LIII 2:383–398
14. Lin X, Ballmann J (1995) A numerical scheme for axisymmetric elastic waves in solids. Wave Motion 21:115–126

15. Lin X, Ballmann J (1995) Numerical modelling of elastic-plastic waves in transversely isotropic composite materials. ZAMM 75:267–268
16. Lin X, Ballmann J (1995) Elastic–plastic waves in cracked solids under plane stress. In: Wegner JL, Norwood FR (eds) Nonlinear waves in solids. ASME book No. AMR, vol 137, pp 155–160
17. Lin X, Ballmann J (1994) Numerical Modelling of Elastic–Plastic Deformation at Crack Tips in Composite Material under Stress Wave Loading. Journal de Physique III, 4:S. C8-53-58
18. Lubliner J (1990) Plasticity theory. Macmillan Publishing Company, New York
19. Morton KW, Richtmyer RD (1995) Difference methods for initial-value problems. Krieger Publishing Company, New York
20. Nowacki WK (1978) Stress waves in non-elastic solids. Oxford/Pergamon Press, New York
21. Pember RB, Bell JB, Collela P, Crutchfield WY, Welcome ML (1995) An adaptive Cartesian grid method for unsteady compressive flow in irrgular regions. J Comput Phys 120:278–304
22. Prandtl L (1924) Proceedings of the First International Congress for Applied Mechanics. Technische Boekhandel en Drukkerij J.Waltman Delft 1:43
23. Quirk JJ (1994) An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies. Comput Fluids 23(1):125–142
24. Serre D (1999) Systems of conservation laws. Cambridge University Press, London
25. Ning Nan, Ting TCT (1969) Plane wave due to combined compressive and shear stresses in half space. J Appl Mech 1:189–197